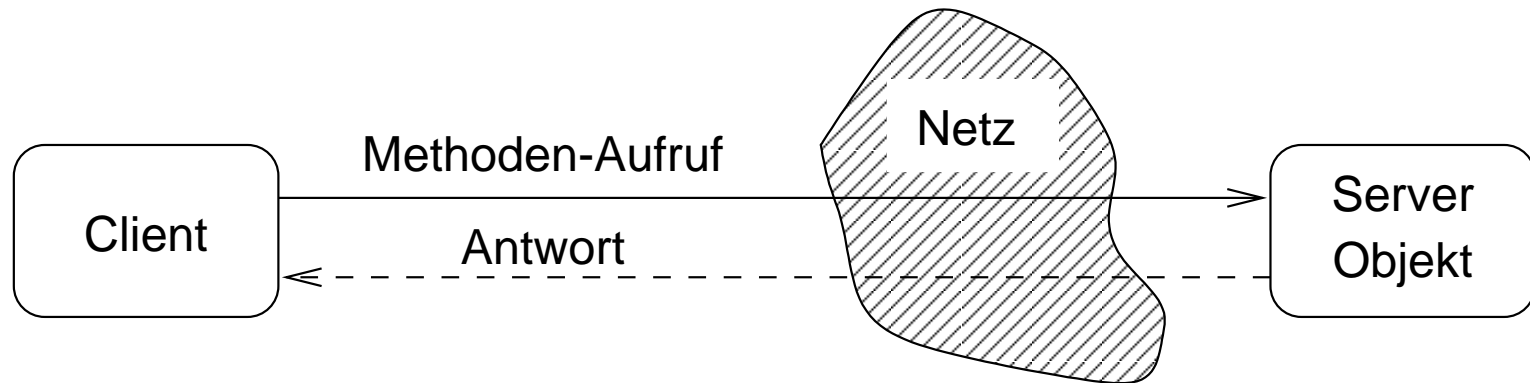


Java RMI, CORBA und Firewalls

Rainer Falk
Lehrstuhl für Datenverarbeitung
TU München
falk@ei.tum.de

- Verteilte Objekte
- Firewalls
- Probleme
- Lösungsmöglichkeiten
- Konkrete Lösungen

Verteilte Objekte



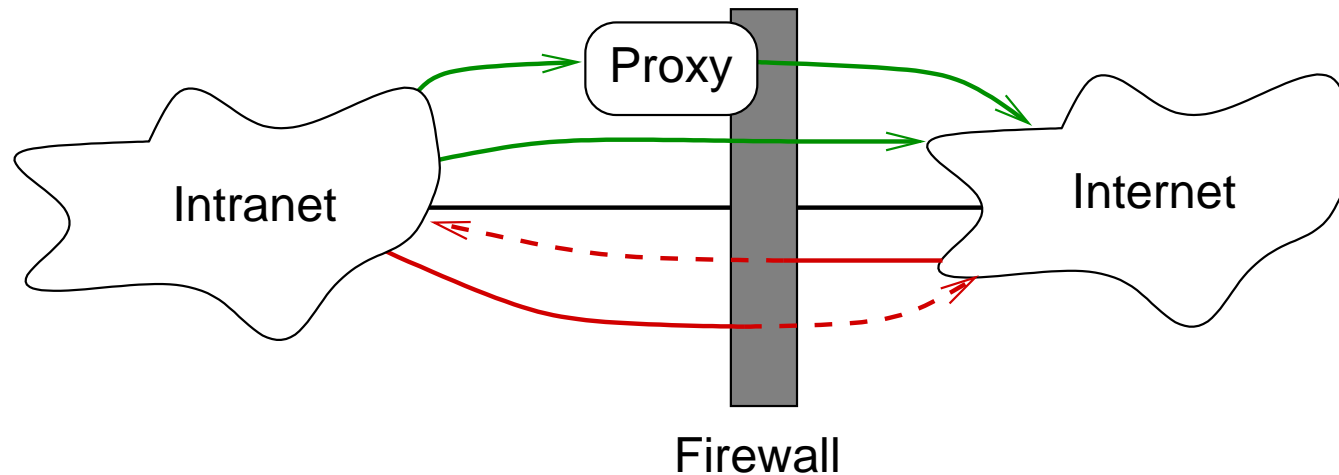
- mehrere Objekte
- auch dynamisch erzeugte
- keine feste Verteilung der Rollen Client und Server (Callback)

Beispiele

Java RMI, CORBA, DCOM, DSOM, ...



Firewall

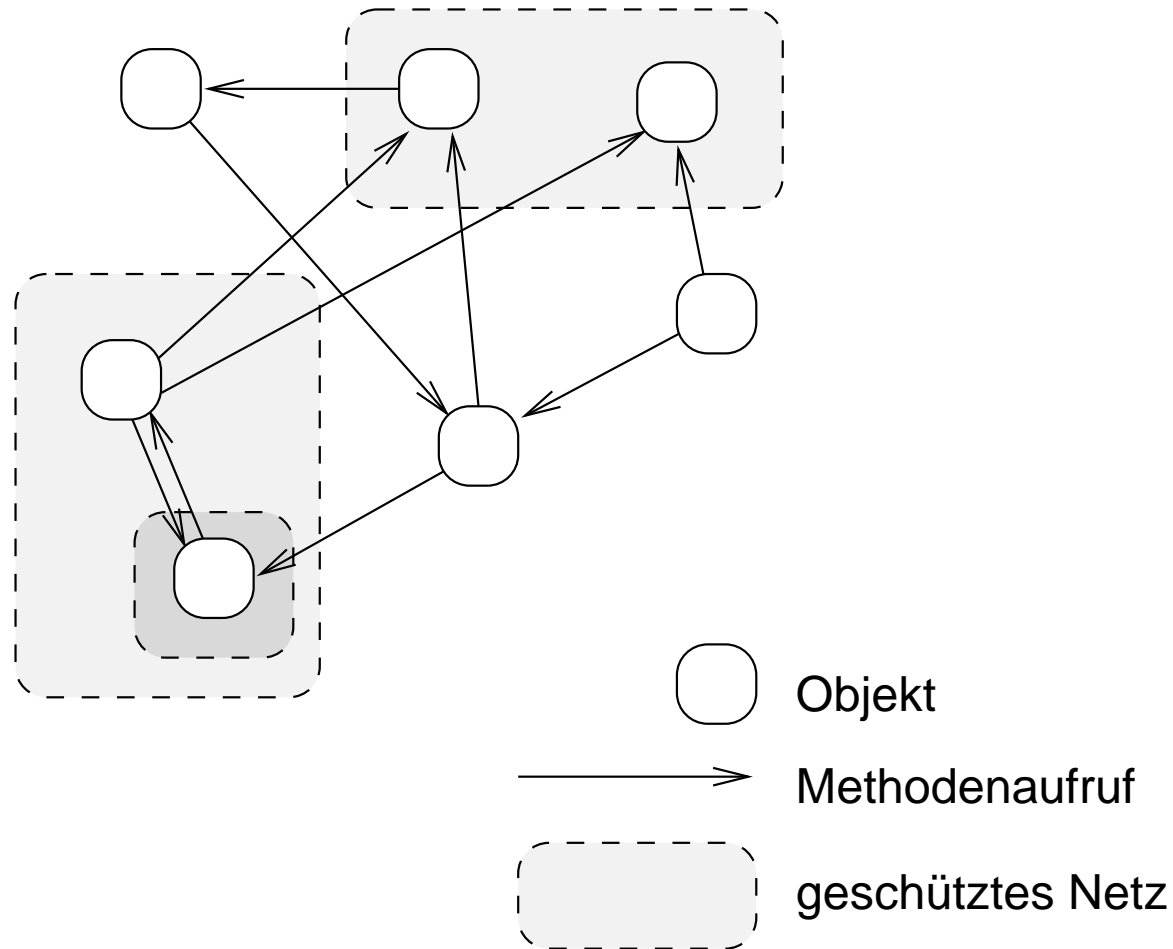


- Überwachung und Logging des Netzverkehrs zwischen zwei Netzen
- nur zugelassener Netzverkehr darf passieren
- oft keine direkte Kommunikation zugelassen (nur über Proxy)
- Verbergen der inneren Netzstruktur

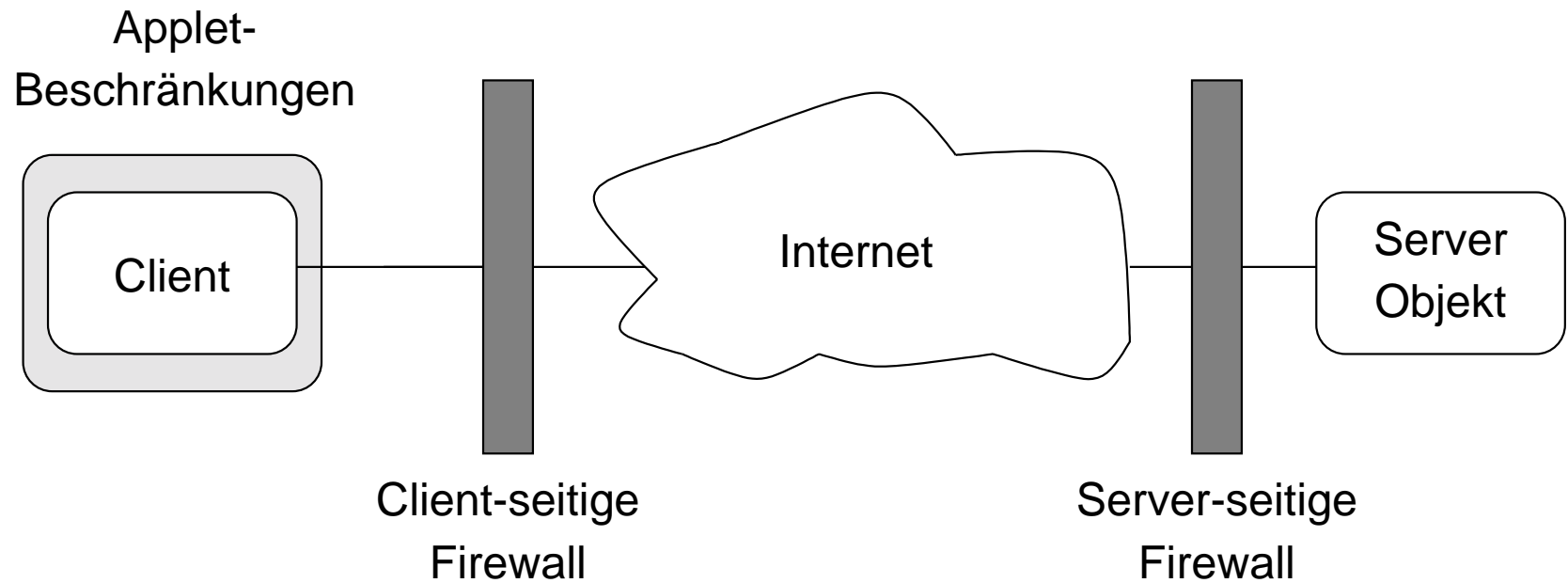
Bausteine: Paket-Filter, Proxies



Verteilte Objekte und Firewalls



Probleme



Server-seitige Firewall

Ansprechen des Zielobjekts

Durchlässigkeit der Firewall abhängig von

- Zielobjekt
- Methode
- Aufrufer

Schwierig für Firewall

- Ortstransparenz
- keine festen Portnummern
- dynamisch erzeugte Objekte (factory/finder objects)
- evtl. keine global gültigen IP-Adressen im Intranet



Client-seitige Firewall

- Überquerung
 - Konfiguration kann angepaßt werden
 - Konfiguration kann nicht angepaßt werden
- Callback-Problematik

Applet-Beschränkungen

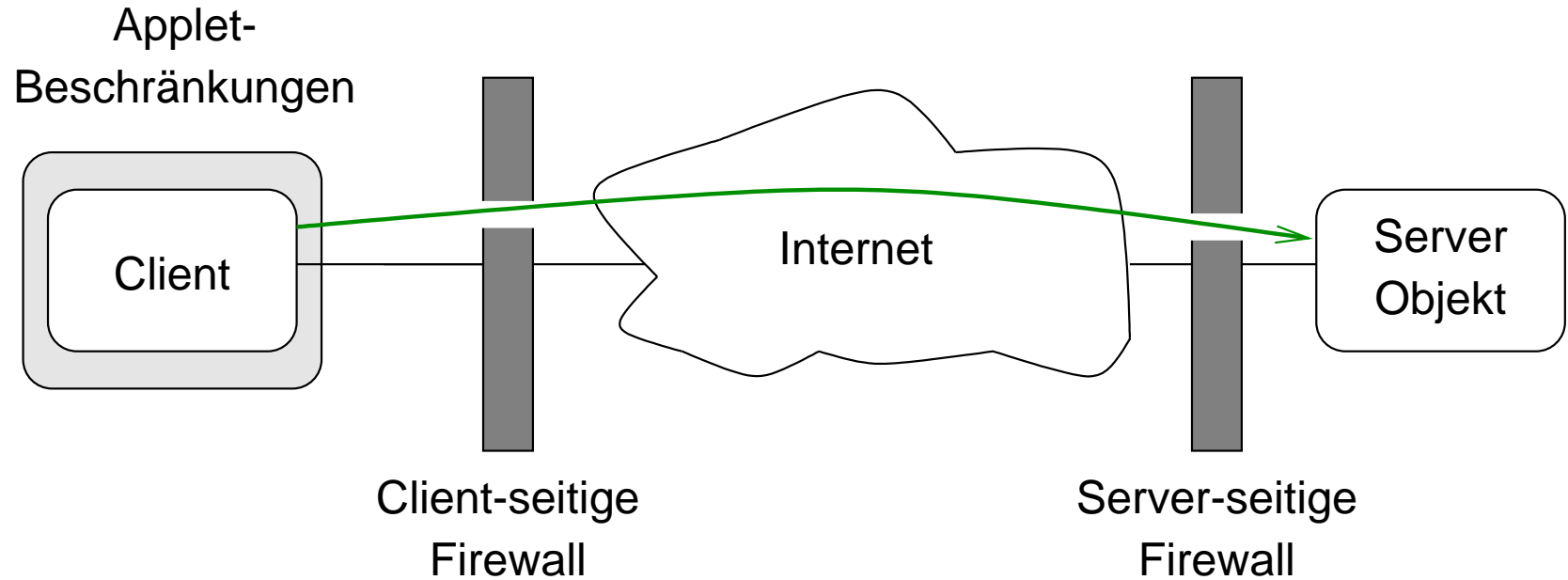
- Verbindungsaufbau nur zu dem Rechner, von dem Applet geladen wurde
- kein Verbindungsaufbau zum Applet

Lösungsmöglichkeiten

- Port öffnen (direkte Verbindung zulassen)
- Tunnelung
- Proxy
 - Server-seitiger Proxy
 - Client-seitiger Proxy



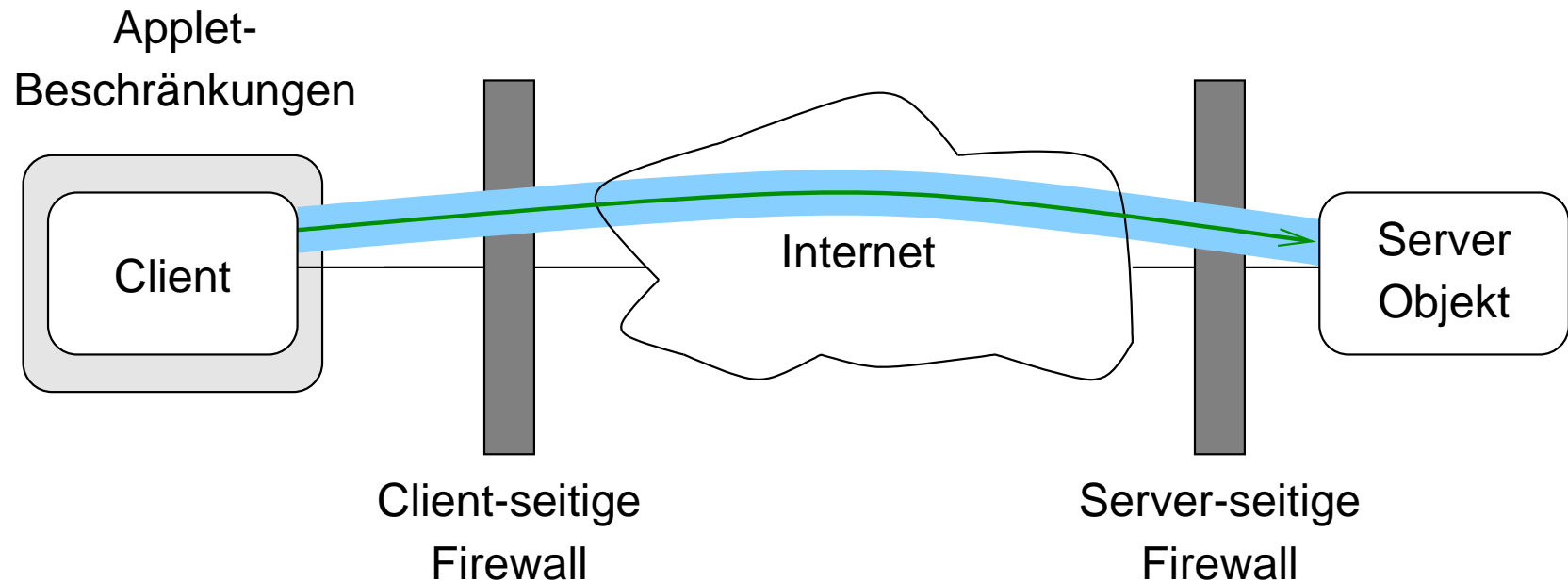
Port öffnen



Probleme

- Ortstransparenz
- dynamische Portnummern \Rightarrow mehr zugelassen als erwünscht
- dynamische Objekte
- Callback

Tunnelung

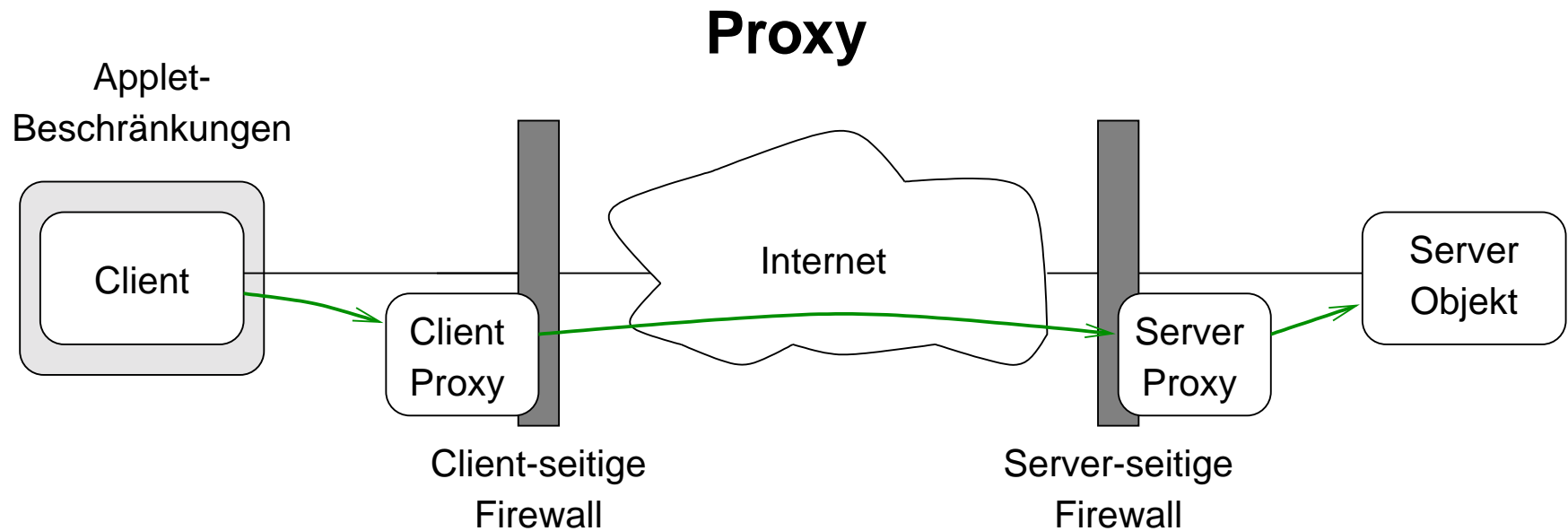


in zugelassenes Protokoll einbetten

Überwinden der Client-Firewall

Protokoll: z.B. HTTP

Client-Firewall kann nutzbare Zielports einschränken



- Netzverkehr nur zu Proxy (feste Portnummer, Adresse)
- Zugriffskontrolle
- Tunnelung
- Umgehung der Applet-Beschränkungen

Konkrete Lösungen

- Java RMI
- CORBA-Proxy
 - Visigenic Gatekeeper
 - IONA Wonderwall
- OMG Request for Proposals zu CORBA Firewall Traversal



Firewall-Problematik berücksichtigt

1. Zielobjekt direkt ansprechen
2. einbetten in HTTP-POST-Anfrage
(direkt an Zielobjekt)
3. einbetten in HTTP-POST-Anfrage
(an WWW-Server: CGI-Skript `/cgi-bin/java-rmi`)

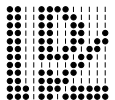
Einschränkungen

- bei Einbettung in HTTP kein Callback



CORBA-Proxy: Wonderwall, Gatekeeper

- Einschränkung der verwendeten Ports
- Web-Server-Funktionalität
- HTTP-Tunnelung (mit jeweils passendem Client-ORB)
- Adressierung des Proxy
 - durch speziellen Client-ORB
 - Wonderwall: Proxifizierung der Objektreferenzen
- jeweils proprietäre Verfahren für Callbacks
 - nicht bei HTTP-Tunnelung
 - nur von Server-Objekt



CORBA-Proxy: Wonderwall, Gatekeeper (2)

IONA Wonderwall

- Zugriffskontrolle auf Objekt-/Methodenebene
- Logging
- auch ORBs anderer Hersteller (dann aufwendiger zu konfigurieren)
- bei Factory/Finder-Objekten
 - alle Objekte auf Rechner ansprechbar
 - Proxifizierung der Objektreferenzen

Visigenic Gatekeeper

- Umgehung der Applet-Beschränkungen bei VisiBroker-Applets
- bei SSL nicht möglich: Web-Server, HTTP-Tunneling



OMG Request for Proposals

Proxy auf Server-Seite

- Erweiterung der Objektreferenzen (IOR) um Information zur Überquerung von Firewalls (auch mehrere)
- Bidirektionale Nutzung von GIOP (Callback)
Kann selbst Sicherheitsproblem sein
- Schnittstelle zur Kommunikation mit CORBA-Proxy
Proxifizierung bei dynamisch erzeugten Objekten

Nicht

- Überquerung von Client-Firewall
Sache des ORB auf Client-Seite, nicht wesentlich für Interoperabilität
- Einbettung in andere Protokolle



Zusammenfassung

- Verteilte Objekte können über Firewalls hinweg angesprochen werden
- Einschränkungen müssen beachtet werden
 - dynamische Objekte (factory/finder object)
 - Callback
 - Anzahl beteiligter Objekte
 - Verzicht auf Ortstransparenz
 - Bindung an speziellen ORB

